

## Durham Research Online

---

### Deposited in DRO:

26 April 2017

### Version of attached file:

Published Version

### Peer-review status of attached file:

Peer-reviewed

### Citation for published item:

Aslett, L. J. M. and Nagapetyan, T. and Vollmer, S. J. (2017) 'Multilevel Monte Carlo for Reliability Theory.', Reliability engineering system safety., 165 . pp. 188-196.

### Further information on publisher's website:

<https://doi.org/10.1016/j.ress.2017.03.003>

### Publisher's copyright statement:

© 2017 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

### Additional information:

---

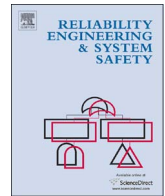
## Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.



# Multilevel Monte Carlo for Reliability Theory

Louis J.M. Aslett\*, Tigran Nagapetyan, Sebastian J. Vollmer

Department of Statistics, University of Oxford, 24–29 St Giles', Oxford OX1 3LB, United Kingdom



## ARTICLE INFO

### Keywords:

Reliability theory  
Multilevel Monte Carlo  
Cut sets  
System lifetime estimation

## ABSTRACT

As the size of engineered systems grows, problems in reliability theory can become computationally challenging, often due to the combinatorial growth in the number of cut sets. In this paper we demonstrate how Multilevel Monte Carlo (MLMC) — a simulation approach which is typically used for stochastic differential equation models — can be applied in reliability problems by carefully controlling the bias-variance tradeoff in approximating large system behaviour. In this first exposition of MLMC methods in reliability problems we address the canonical problem of estimating the expectation of a functional of system lifetime for non-repairable and repairable components, demonstrating the computational advantages compared to classical Monte Carlo methods. The difference in computational complexity can be orders of magnitude for very large or complicated system structures, or where the desired precision is lower.

## 1. Introduction

It can prove to be computationally intractable to perform classical reliability analysis of very large engineered systems when the number of cut (path) sets grows combinatorially. It is well understood that working instead with subsets of the cut (path) sets or bounding structural designs can provide probability bounds in many reliability problems [4], but such bounds can be crude or may not be well characterised at all.

Evaluation of the reliability of engineered systems is a crucial part of system design and often scenario planning may involve repeated evaluation of the reliability for changing system configurations or component types meaning rapid simulation is highly desirable. For simplicity of exposition we herein consider the canonical problem of estimating the expectation of a functional of system lifetime both with and without a component repair process, showing the approach developed is easily generalised to other reliability problems which depend on cut (path) sets for the analysis.

In the case of static reliability analysis, there are many methods aside from Monte Carlo simulation using the cut (path) sets, including Sum of Disjoint Products (SDP) methods [22,30,27] and methods based on Binary Decision Diagrams (BDD) [25] or multistate BDD extensions [29]. On the other hand, these approaches are less prevalent in dynamic reliability problems where there are component dependencies, for example through system shocks, repair or maintenance programmes, and cascading failures among others. There have been recent developments in dynamic fault trees [20,26,21] which apply

where event sequence ordering influences the reliability, including repairable systems [19]. When there are arbitrary dependencies, the most generally applicable approach is direct Monte Carlo simulation (e.g. [7]), so that acceleration of Monte Carlo techniques is important to address a broad range of the most complex reliability scenarios. Monte Carlo acceleration through importance sampling [15], or the use of control variates [28] have been suggested in the context of reliability estimation, but they are either restricted to the static case and require regular updates and sorting of all the cut sets (as for [15]), or could be combined with the MLMC paradigm (as for [28]).

Indeed, also note that interest may not be in the reliability at a particular fixed mission time, but instead in: some expectation of a functional of system lifetime; or in ascertaining a quantile of system lifetime (i.e. the time to which one is 99.9% certain the system will survive); or in estimation of the entire system lifetime distribution. In these situations Monte Carlo methods are typically the only tractable approach.

Multilevel Monte Carlo (MLMC) methods — pioneered by Heinrich [14] and Giles [10] — are now standard for estimation of expectations of functionals of processes defined by stochastic differential equations (SDEs). However, the MLMC approach is in fact a general paradigm for accelerating any Monte Carlo based method (whether standard, importance sampling, etc), if one can link the accuracy of the estimator with the complexity of generating a sample, while at the same time controlling the variance of the difference for approximations with different accuracy. The main contribution in this paper is development of a Multilevel Monte Carlo (MLMC) approach to

\* Corresponding author.

E-mail addresses: [aslett@stats.ox.ac.uk](mailto:aslett@stats.ox.ac.uk) (L.J.M. Aslett), [nagapetyan@stats.ox.ac.uk](mailto:nagapetyan@stats.ox.ac.uk) (T. Nagapetyan).

reliability problems. In this way, we show how any reliability problem using Monte Carlo simulation over cut (path) sets can be substantially accelerated, extending the size of systems and complexity of dependencies which are within reach for reliability evaluation.

In Section 2, we recap the traditional cut set method of simulating system lifetimes which does not scale well to large systems even when the cut sets are known. This motivates the approach taken in this work. In Section 3 we recap standard Monte Carlo theory and set out the error and computational cost metrics which will enable comparison with our MLMC based approach. The fundamental MLMC methodology and our adaptation to the reliability setting then follow in Section 4, before numerical results demonstrating the kind of substantial computational improvements which can be achieved are covered in Section 5.6.

## 2. Simulating system lifetimes

Consider a coherent system with  $n$  components. Let  $x_1(t), \dots, x_n(t)$  denote the operational status (1= working, 0= failed) of the components at time  $t$  and consider the random variable for the lifetime of component  $c$  to be  $T_c \sim F_c(t)$ , where  $F_c(\cdot)$  are positively supported lifetime distributions which are not necessarily independent or identical. We will depict a system as an undirected network comprising a set of nodes (vertices)  $S$ , and a set of edges  $E$ , where nodes are considered unreliable and edges are perfectly reliable (note that any setting with imperfect edge reliability can be easily transformed to a corresponding representation where they are perfectly reliable [2]).

The system is considered to be functional as long as there is a path from left to right which passes only through functioning nodes, see Fig. 1. This is usually represented mathematically by the structure function,  $\phi: \{0, 1\}^n \rightarrow \{0, 1\}$ , which maps component status to system status.

Herein, our focus is on an equivalent means of evaluation based on cut sets. A set of components,  $C$ , is said to be a cut set of the system if the system is failed whenever all the components in  $C$  are failed. A cut set is said to be a minimal cut set if no subset of it is also a cut set. Then, the set of all minimal cut sets,  $C$ , characterises the operational state of a system completely and is equivalent to knowledge of the structure function [6]. In addition to the cut sets characterising the operational state of the system given the binary operational state of the components, they also immediately provide the system failure time if the individual component failure times are known [5]:

$$T_S = f_S(T_1, \dots, T_n) := \min_{C \in C} \left\{ \max_{c \in C} \{T_c\} \right\}.$$

Thus, the failure time for the system depends on the system structure and the failure time distributions for each node.

The traditional approach to estimating the expectation of a functional of the lifetime of a system given the lifetime distributions of the components is to perform a simple Monte Carlo simulation. That is,

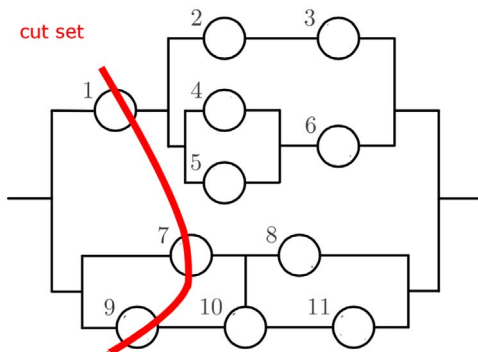


Fig. 1. A sample network with a minimal cut set.

$$\mathbb{E}[g(T_S)] = \frac{1}{N} \sum_{i=1}^N g(f_S(t_1^{(i)}, \dots, t_n^{(i)})) \text{ where } t_j^{(i)} \sim F_j(\cdot).$$

The overall runtime for this approach depends on three quantities:

1. Variance of the estimator. Due to the random nature of component failure times, the estimator is a random variable: higher variance estimators will require more iterations to achieve an accurate estimate;
2. Target accuracy of the estimate. Naturally, the higher the desired accuracy, the longer the algorithm will take due to more iterations being required;
3. Number of cut sets. As the system size grows, the number of cut sets has a combinatorial growth, making the approach impractical for very large systems.

Less brute force approaches are possible with the restrictive assumption of iid components by making use of the system signature [18,23]. More recent work on the survival signature [8] has generalised the signature to multiple types of component, with the weaker assumption of exchangeability between components. However, if a large number of the components are of different types or there are highly dependent failures, then the survival signature will also grow exponentially in complexity. It can also accommodate a repair process [9] through expression as a new component type, though this increases the complexity if too many repairs occur. Hence, in this work, we first address the most general possible setting in which any form of component lifetime and dependence structure is allowed, requiring only knowledge of component lifetimes and the cut sets of the system. However, note that it should be possible to specialise this approach to work with the survival signature which we hope to address in future research.

## 3. Monte Carlo algorithms

To simplify presentation, hereinafter we only consider estimating expected failure time directly, rather than some functional of the failure time, though this is mostly without loss of generality (see Section 4 for details). Therefore, assume that for a given system  $S$ , we want to estimate the expected failure time.

$$\mathbb{E}T_S = \mathbb{E}f_S(T_1, \dots, T_n).$$

There are many approaches to simulation which may differ in terms of convergence to the true value as well as computational characteristics. In order to compare them, we present some useful cost and error expressions in the following subsection.

### 3.1. Performance measures: error and cost definitions

We start by defining the two main quantities, which will be used throughout this paper to compare methodologies. Given an estimator  $\hat{T}_S$  of the quantity  $\mathbb{E}T_S$ , the Mean Squared Error (MSE) of any Monte Carlo based method is:

$$\text{error} = \mathbb{E}[(\hat{T}_S - \mathbb{E}T_S)^2].$$

The classical decomposition of the MSE yields:

$$\mathbb{E}[(\hat{T}_S - \mathbb{E}T_S)^2] = \mathbb{E}[(\hat{T}_S + \mathbb{E}\hat{T}_S - \mathbb{E}\hat{T}_S - \mathbb{E}T_S)^2] = \mathbb{E}[(\hat{T}_S - \mathbb{E}\hat{T}_S)^2] + (\mathbb{E}\hat{T}_S - \mathbb{E}T_S)^2 \quad (1)$$

where  $(\mathbb{E}\hat{T}_S - \mathbb{E}T_S)^2$  is the squared bias error, while  $\mathbb{E}[(\hat{T}_S - \mathbb{E}\hat{T}_S)^2]$  is the error due to Monte Carlo variance. The first is a systematic error arising from the fact that we might not sample our random variable exactly, but rather use a suitable approximation, while the second error comes from the randomised nature of any Monte Carlo algorithm. For example, in traditional Monte Carlo applications, one samples exactly

so that the first error is zero and only the Monte Carlo variance needs to be treated carefully.

The cost of any Monte Carlo based algorithm is typically taken to be the expected runtime in order to achieve a prescribed accuracy. A more convenient approach for theoretical comparison between different methods is to define

$$\text{cost} = \mathbb{E}(\text{\#random number generations and operations}).$$

We now recap traditional Monte Carlo and then introduce Multilevel Monte Carlo, in each case highlighting their corresponding results for these two measures of performance.

### 3.2. Traditional (or single-level) Monte Carlo algorithm

The traditional Monte Carlo estimator is based on  $N$  replications of simulating the system lifetime, via the minimal cut sets, by simulating the component lifetimes. That is, given system simulations  $\tau_i = f_S(t_1^{(i)}, \dots, t_n^{(i)})$  the traditional Monte Carlo estimator has the form

$$\hat{T}_S = \frac{1}{N} \sum_{i=1}^N f_S(t_1^{(i)}, \dots, t_n^{(i)}). \quad (2)$$

For reasons that will become clear in the sequel, it is useful to refer to this as the single-level Monte Carlo algorithm because it emphasises the relationship to Multilevel Monte Carlo.

This single-level Monte Carlo estimate has variance proportional to  $N^{-1}$ ,

$$\text{Var}(\hat{T}_S) = \text{Var}\left(\frac{1}{N} \sum_{i=1}^N \tau_i\right) = \frac{1}{N^2} \text{Var}\left(\sum_{i=1}^N \tau_i\right) = \frac{1}{N} \text{Var}(\tau_i).$$

The estimator (2) is clearly unbiased, because there is no approximation involved in estimating the failure time, so the error measure introduced earlier only has this second variance term,

$$\text{error}_{\text{MC}} = N^{-1} \text{Var}(\tau_i).$$

Indeed, more generally the well known central limit result for standard Monte Carlo means that:

$$\mathbb{P}\left(|\hat{T}_S - \mathbb{E}T_S| > z \frac{\sqrt{\text{Var}(\tau_i)}}{\sqrt{n}}\right) \approx \mathbb{P}(|Z| > z)$$

for  $Z \sim N(0, 1)$ .

Thus, for a desired level of accuracy  $\varepsilon > 0$  with confidence level  $1 - \alpha$ , we require

$$n = z_{\alpha/2}^2 \text{Var}(\tau_i) \varepsilon^{-2}$$

Monte Carlo simulations, where the quantile  $z_{\alpha/2}$  is chosen to ensure that  $\mathbb{P}(Z > z_{\alpha/2}) = \alpha/2$ .

Naturally  $z_{\alpha/2}$  is a constant for any fixed level of confidence, so the variable compute costs in simulation are

$$\text{cost}_{\text{MC}} = \text{Var}(\tau_i) \cdot \varepsilon^{-2} \cdot \#C, \quad (3)$$

where  $\#C$  denotes the number of minimal cut sets for the system.

## 4. Multilevel Monte Carlo

To simplify presentation we again only consider estimating expected failure time directly, rather than some functional of the failure time. Note that there is no loss of generality, so long as the functional of interest is Lipschitz continuous (or bounded for discrete measures). The most common functional of interest that this would exclude is computing expectations of quantiles. However, this problem can be treated with a smoothing approach, as discussed for the MLMC setting in [13]. In all other cases, the presentation below carries over in the natural fashion.

### 4.1. General MLMC

We first introduce MLMC in generality before specialising this to the reliability problem considered herein. Consider a sequence of estimators  $T_0, T_1, \dots$ , which approximates  $T_L$  with increasing accuracy, but also increasing cost. By linearity of expectation, we have

$$\mathbb{E}(T_L) = \mathbb{E}(T_0) + \sum_{\ell=1}^L \mathbb{E}[T_\ell - T_{\ell-1}],$$

and therefore we can use the following unbiased estimator for  $\mathbb{E}[T_L]$ ,

$$\frac{1}{N_0} \sum_{n=1}^{N_0} T_0^{(0,n)} + \sum_{\ell=1}^L \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} (T_\ell^{(\ell,n)} - T_{\ell-1}^{(\ell,n)}) \right\}$$

The inclusion of the level  $\ell$  in the superscript  $(\ell, n)$  indicates that the samples used at each level of correction are independent, but crucially note that the differences themselves use common samples. Note the terminology ‘correction’ arises from the fact that each  $T_\ell$  is generally not an unbiased estimate any more.

Let  $V_0$  and  $\text{cost}_0$  be the variance and the expected cost of one sample of  $T_0$ , and let  $V_\ell$ ,  $\text{cost}_\ell$  be the variance and expected cost of one sample of  $T_\ell - T_{\ell-1}$ . Then the overall expected cost and variance of the multilevel estimator are  $\sum_{\ell=0}^L N_\ell \cdot \text{cost}_\ell$  and  $\sum_{\ell=0}^L N_\ell^{-1} V_\ell$ , respectively.

More generally, this means that provided the product  $V_\ell \cdot \text{cost}_\ell$  decreases with  $\ell$  (i.e. the cost increases with level slower than the variance decreases), then one can achieve significant computational savings, which can be formalised in Theorem 1 from [10].

**Theorem 1.** Let  $T_S$  denote a random variable, and let  $T_\ell$  denote the corresponding level  $\ell$  numerical approximation.

If there exist independent estimators  $Y_\ell$  based on  $N_\ell$  Monte Carlo samples, and positive constants  $\alpha, \beta, \gamma, c_1, c_2, c_3$  such that  $\alpha \geq \frac{1}{2} \min(\beta, \gamma)$  and

1.  $|\mathbb{E}(T_\ell - T_S)| \leq c_1 2^{-\alpha \ell}$
2.  $\mathbb{E}(Y_\ell) = \begin{cases} \mathbb{E}(T_0), & \ell = 0 \\ \mathbb{E}(T_\ell - T_{\ell-1}), & \ell > 0 \end{cases}$
3.  $\text{Var}(Y_\ell) \leq c_2 N_\ell^{-1} 2^{-\beta \ell}$
4.  $\text{cost}_\ell \leq c_3 2^{\gamma \ell}$ , where  $\text{cost}_\ell$  is the expected computational complexity of  $Y_\ell$

then there exists a positive constant  $c_4$  such that for any  $\varepsilon < e^{-1}$  there are values  $L$  and  $N_\ell$  for which the multilevel estimator

$$Y = \sum_{\ell=0}^L Y_\ell,$$

has a mean-square-error with bound

$$\text{MSE} \equiv \mathbb{E}[(Y - \mathbb{E}[T_S])^2] < \varepsilon^2$$

with a computational complexity  $C$  with bound

$$\text{cost}_{\text{MLMC}} \leq \begin{cases} c_4 \varepsilon^{-2}, & \beta > \gamma, \\ c_4 \varepsilon^{-2} (\log \varepsilon)^2, & \beta = \gamma, \\ c_4 \varepsilon^{-2 - (\gamma - \beta)/\alpha}, & \beta < \gamma. \end{cases}$$

**Remark 2.** We will informally illustrate the idea behind MLMC on a simple example with just two levels. Consider just two approximations  $T_k$  and  $T_L$ , where  $k < L$ , with sample costs  $\text{cost}_k < \text{cost}_L$ . It is clear, that the cost of one sample for  $T_L - T_k$  is roughly  $\text{cost}_L$ . Now assume, that

$$V_1 = \text{Var}T_k \approx \text{Var}T_L, \text{ and } V_2 = \text{Var}(T_L - T_k),$$

where  $V_2 < V_1$ . Then we have

$$\begin{aligned}\mathbb{E}T_L &= \mathbb{E}T_k + \mathbb{E}(T_L - T_k) \Rightarrow \hat{T} = \frac{1}{N} \sum_{n=1}^N T_L^{(2,n)} \\ &\approx \bar{T} = \frac{1}{N_1} \sum_{i=1}^{N_1} T_k^{(1,i)} + \frac{1}{N_2} \sum_{j=1}^{N_2} (T_L^{(2,j)} - T_k^{(2,j)}).\end{aligned}$$

We see that the overall cost of Monte Carlo estimators, according to (3), can be expressed as

$$\begin{aligned}\text{cost}(\hat{T}) &= \varepsilon^{-2} \cdot V_1 \cdot \text{cost}_L \\ \text{cost}(\bar{T}) &= \varepsilon^{-2} \cdot (\text{cost}_k \cdot V_1 + \text{cost}_L \cdot V_2),\end{aligned}$$

which gives us a condition

$$\text{cost}(\hat{T}) > \text{cost}(\bar{T}) \Rightarrow 1 > \frac{\text{cost}_k}{\text{cost}_L} + \frac{V_2}{V_1}.$$

In other words, provided there exists a good coupling between estimators  $T_k$  and  $T_L$ , we have reduced computational cost even for two levels. The two-level Monte Carlo method in the context of Monte Carlo path simulation has been suggested and analysed in [16].

**Remark 3.** Multilevel provides the greatest benefit when  $\beta \geq \gamma$ , because this is the case for which we get the best asymptotic performance.  $\gamma$  represents the parameter for the exponential increase of the cost of producing a sample, while  $\beta$  corresponds to the parameter for the exponential decay in the variance of the sample at a given level. There are 3 cases:

$\beta < \gamma$ . The number of samples required by the MLMC estimator decays at a slower rate than the increase of the sampling cost at each level. In this case the overall cost is proportional to the cost of the last level.

$\beta = \gamma$ . This is most common in practice [12]. The decay of the variance is balanced with the increase of the cost, therefore the contribution to the overall cost is the same from all the levels.

$\beta > \gamma$ . In this most desirable case, the overall cost is dominated by level  $\ell = 0$ , since consecutive levels will have a decaying contribution to the cost.

When not available analytically, estimation of  $\alpha$ ,  $\beta$  and  $\gamma$  is usually done by empirically regressing using diagnostic quantities in the manner demonstrated in Sections 5 and 6 for our examples.

**Remark 4.** Multilevel Monte Carlo became popular after the seminal work of Mike Giles [10] for estimating expectations of functionals  $\mathbb{E}(f(X_t))$ , where  $X_t$  is the solution of a stochastic differential equation. In the general Multilevel Monte Carlo path simulation setting,  $T_\ell$  from Theorem 1 is the functional value, evaluated via an approximation arising from a discretisation method, e.g. the Euler-Maruyama method [17].

#### 4.2. MLMC for system reliability

Theorem 1 suggests that one may want to try getting a coarser Monte Carlo estimate of the system lifetime, perhaps by considering only a subset of the collection of minimal cut sets.

$$C' \subset C \Rightarrow \min_{C \in C'} \left\{ \max_{i \in C} \{t_i\} \right\} = T'_S \geq T_S = \min_{C \in C} \left\{ \max_{i \in C} \{t_i\} \right\}.$$

On its own  $T'_S$  is a biased estimator, so although a traditional single-level Monte Carlo estimator based on it may have lower computational cost, it will have increased MSE because the first error term in (1) can no longer be ignored. However, by using this coarse estimate inside an MLMC approach, we aim to improve the overall performance. To this end we introduce the sequence of estimators  $T_0, \dots, T_L$  based on a nested sequence of minimal cut sets,  $C_0 \subset \dots \subset C_L = C$ . Note that here  $T_L \equiv T_S$ , which is not typically true

in a general MLMC setting.

The crucial ingredient is the finite telescopic sum

$$\mathbb{E}(T_S) \equiv \mathbb{E}(T_0) + \sum_{\ell=1}^L \mathbb{E}(T_\ell - T_{\ell-1}) = \sum_{\ell=0}^L \mathbb{E}(Y_\ell)$$

As described above, we independently estimate each term, and within each term,  $T_\ell$  and  $T_{\ell-1}$  use the same random component simulations:

$$\mathbb{E}(Y_\ell) \approx N_\ell^{-1} \sum_{j=1}^{N_\ell} (\tau_\ell^{(j)} - \tau_{\ell-1}^{(j)})$$

with each level having cost being bounded from above by  $c \cdot \text{Var}(Y_\ell) \cdot \#C_\ell$ . Here  $c$  is a constant independent of  $\ell$  and the desired target accuracy. We choose  $\#C_\ell$  – the number of minimal cut sets at level  $\ell$  – to be a proxy for the upper bound on the cost of each level, because for a fixed system the number of elements in each minimal cut set is independent of the target accuracy. In other words, as we double the number of minimal cut sets in each level, their number is a straightforward way to construct a meaningful and easy upper bound for the cost of one sample.

Thus, the overall MLMC variance is

$$\text{Var}\left(\sum_{\ell=0}^L Y_\ell\right) = \sum_{\ell=0}^L N_\ell^{-1} \text{Var}(Y_\ell)$$

at a cost of  $\sum_{\ell=0}^L N_\ell \cdot \#C_\ell$ . Therefore, given a fixed target accuracy (variance), if we choose a sample size  $N_\ell \approx \sqrt{\text{Var}(Y_\ell) / \#C_\ell}$  on each level, this will minimise the computational cost. That is, for a desired accuracy  $\varepsilon > 0$ , the overall cost is:

$$\text{cost}_{\text{MLMC}} = \sum_{\ell=0}^L N_\ell \cdot \#C_\ell = \varepsilon^{-2} \left( \sum_{\ell=0}^L \sqrt{\text{Var}(Y_\ell) \cdot \#C_\ell} \right)^2 \quad (4)$$

Recall that,

$$\text{cost}_{\text{MC}} = \text{Var}(\tau_1) \cdot \varepsilon^{-2} \cdot \#C,$$

This means that:

1. If we have a good coupling between the approximations, or equivalently  $\text{Var}(Y_\ell)$  decays rapidly, then we can achieve considerable savings compared to the single-level Monte Carlo algorithm.
2. Additional savings are possible if we do not calculate all the levels  $Y_\ell$ , but rather stop the algorithm early. This introduces a (small) bias, but substantially decreases the overall computational cost. As long as the bias is quantified – and when combined with the estimator variance gives a MSE (1) below our target accuracy – then we can still solve the original problem at much lower cost.

Our proposed application to reliability involves a nested sequence of minimal cut sets providing improving accuracy:

$$C_0 \subset \dots \subset C_L = C,$$

so the possible gain from stopping early depends on the way the minimal cut sets are chosen at each level.

##### 4.2.1. Level selection algorithm

The first point to note is that existing Multilevel Monte Carlo literature has shown that anything less than geometric decay in the cost of computation at each level leads to suboptimal gains, see [12]. Therefore, we pre-specify that level  $\ell$  contain  $\lceil \#C/2^{L-\ell} \rceil$  minimal cut sets. The levels will be grown from  $\ell = 0$  up, adding in those minimal cut sets which are in some sense optimal for the next level. Thus, we specify level selection in an inductive fashion. Note, we initially ignore the possibility of repair for simplicity of presentation and address the changes involved to accommodate repair in Section 6.2, when we demonstrate MLMC for repairable components.



### Selection of level $\ell = 0$

Level 0 will be simulated most frequently since it is the lowest cost. Therefore, optimal choice of this level is straightforward: it should contain the minimal cut sets which provide the best approximation to  $T_0$ . That is, we wish to assign to level 0 the minimal cut sets which have smallest expected failure time, since these will most frequently be the causes of system failure. To achieve this, we propose an initial highly crude estimate by performing a pilot standard Monte Carlo simulation of  $N'$  lifetimes of each component in the system, using these to generate  $N'$  realisations of the failure time associated with each minimal cut set,

$$\eta_i = \frac{1}{N'} \sum_{j=1}^{N'} \max_{c \in C_i} \{t_c^{(j)}\}, \quad \forall C_i \in C. \quad (5)$$

The cut sets corresponding to the smallest  $\lceil \#C/2^L \rceil$  of these  $\eta_i$  are then chosen to form  $C_0$ .

### Selection of level $\ell > 0$

Given that we have chosen levels  $0, \dots, \ell - 1$  already (that is  $C_0 \subset \dots \subset C_{\ell-1}$  are now fixed), we need to select which cut sets to add from  $C_{\text{trial}} = C \setminus C_{\ell-1}$ . To maximise the performance of MLMC we would like to select the cut sets such that

$$\mathbb{E}[T_{\ell-1} - T_\ell] \rightarrow \max.$$

In other words, choose

$$C_\ell = \arg \max_{C_\ell \subseteq C} \mathbb{E}[T_{\ell-1} - T_\ell],$$

s.t.  $C_{\ell-1} \subset C_\ell, \#C_\ell = \lceil \#C/2^{L-\ell} \rceil$

so that the contribution from each level is as large as possible in the smallest levels, leading to a rapid decay in the size of the contribution in each level and hence the possibility of terminating the algorithm early. In particular, note that if  $\sum_{\ell=k}^L \mathbb{E}(T_\ell - T_{\ell-1}) \ll \varepsilon$ , then levels  $k, \dots, L$  need not be simulated at all, so that ensuring all large differences occur in early levels is highly desirable.

Notice that:

$$\mathbb{E}[T_{\ell-1} - T_\ell] \leq \mathbb{E}\left[T_{\ell-1} - \min_{C \in C} \left\{T_{\ell-1}, \max_{c \in C} \{t_c\}\right\}\right],$$

for any  $C \in C_{\text{trial}}$ . So our choice for sorting cut sets is motivated by the minimisation of the upper bound for the increments at each level, which we can implement for any level  $\ell$  in a simple way:

- use the  $N'$  samples and calculated failure times for all cut sets used in selecting  $\ell = 0$ ,
- calculate the following estimates:

$$\delta_k = \frac{1}{N'} \sum_{j=1}^{N'} \left[ T_{\ell-1} - \min_{C \in C_k} \left\{ T_{\ell-1}, \max_{c \in C_k} \{t_c^{(j)}\} \right\} \right],$$

for each  $C_k \in C_{\text{trial}}$ .

- Sort  $\delta_k$  in a descending order and add cut sets corresponding to the largest values for  $\delta_k$  to  $C_\ell$  until  $\#C_\ell = \lceil \#C/2^{L-\ell} \rceil$ .

This choice of number of minimal cut sets on each level guarantees an exponential increase in the cost, corresponding to  $\gamma = 1$  in [Theorem 1](#).

### 4.3. Full Multilevel Monte Carlo algorithm for reliability

One of the key features of the Multilevel Monte Carlo algorithm is its ability to naturally provide stopping criteria for an optimal selection of the number of levels  $\hat{L}$  to actually simulate, which we illustrate now along with the full description of the algorithm. For more advanced approaches to implementation of Multilevel Monte Carlo we refer to [\[11,3,12\]](#).

According to [Theorem 1](#) and the first assumption in it, we have asymptotically as  $\ell \rightarrow \infty$

$$\mathbb{E}(Y_\ell) \approx \mathbb{E}(T_S - \hat{T}_\ell),$$

so that a natural stopping criteria is to choose  $\hat{L}$  minimal such that  $|Y_{\hat{L}}| \leq \varepsilon/2$ .

Input: Required accuracy  $\varepsilon$  and level specification as per [§4.2](#).

1. Set the initial number of levels to  $\hat{L}:=2$ . In order to define the optimal number of samples at each level we need to estimate the variances on levels  $\ell = 0, 1, 2$
2. Compute  $N_\ell:=100$  samples on levels  $\ell = 0, 1, 2$
3. Estimate  $\text{Var}(Y_\ell)$  and update  $N_\ell$  for each level  $\ell = 0, \dots, \hat{L}$ :

$$N_\ell := \max\{N_\ell, \hat{N}_\ell\}, \text{ where}$$

$$\hat{N}_\ell = \left\lceil 4 \cdot \varepsilon^{-2} \cdot \sqrt{\text{Var}(Y_\ell) 2^{-\ell}} \cdot \sum_{k=0}^{\hat{L}} \sqrt{\text{Var}(Y_k) \cdot 2^k} \right\rceil$$

We take the maximum as it may happen that the numerical variance was initially overestimated and so more simulations were performed than necessary. If  $N_\ell$  has increased less than 1% on all levels, then skip to step 5.

4. Compute the additional number of samples on each level  $\ell = 0, \dots, \hat{L}$  and return to step 3.
5. Upon reaching this step we have converged in terms of MSE due to the variance. Next we test whether the bias error term is sufficiently small to terminate, or whether more levels and simulations are required. If  $|Y_{\hat{L}}| \geq \varepsilon$ :

Then: set  $\hat{L} := \hat{L} + 1$ ,  $\text{Var}(\hat{Y}_{\hat{L}}) = \text{Var}(Y_{\hat{L}-1})/2$  and return to step 3;

Else: Terminate algorithm returning  $\sum_{\ell=0}^{\hat{L}} Y_\ell$  as the estimate.

There are two extreme cases to bear in mind. In the first case, we have only a few minimal cut sets (or even only one), which influence the failure time. This case is treated with the initial choice of the cut sets and selection as prescribed in [§4.2](#) should ensure the minimal number of levels is simulated. The second case is when all the cut sets have similar ‘weight’ in determining the failure time, such as with a fully connected system with independent and identically distributed failure times for all components. This case is treated with the doubling of the number of cut sets with respect to the previous level, which again assures the mean and variance decay between the levels.

**Remark 5.** The Multilevel paradigm, with some slight modifications, also allows construction of efficient numerical algorithms (see [\[13\]](#)) for estimating the distribution function on a compact interval in a uniform norm. More specifically, one can construct an algorithm which allows estimates in the norm

$$\left( \mathbb{E} \sup_{t \in [t_0, t_1]} |\mathbb{P}(T_S < t) - \mathbb{P}(\hat{T}_S < t)|^2 \right)^{1/2},$$

which guarantees the uniformity in the error for numerical estimates. This is the only Monte Carlo based approach whose estimates are functions, rather than finite dimensional entities, which has uniform norm as a measure of accuracy.

## 5. Numerical experiments, no repair

### 5.1. Systems and component reliability distributions

We generated many random systems to test the MLMC reliability method proposed hereinbefore. These random systems are generated by starting from the trivial one component system and with fixed probabilities either:

- replacing a component with two components in series;
- replacing a component with two components in parallel;
- selecting two edges and inserting a ‘bridging’ component.

This allowed us to generate a wide range of different systems and in particular an increasing sequence of related systems with varying numbers of component.

For all systems we consider three test cases, where the components have Weibull distributed lifetimes with shape parameter  $k = 0.5, 1$  or  $3$  and where the scale is chosen uniformly at random on an interval  $[2, 10]$ . Variety in shape parameters corresponds to different applications in industry (see e.g. [24]). The shape parameter has a substantial effect on the corresponding density function.

## 5.2. Numerical results

We ran our algorithm 100 times for systems of different sizes with independent but differently distributed components, whose reliability is described above. In each case we considered fixed target accuracies of  $\varepsilon = 2^{-4}$  and  $\varepsilon = 2^{-7}$ , and computed the cost gains achieved for these fixed accuracies.

### 5.2.1. Shape parameter $k=0.5$

The left top and bottom plots in Fig. 2 show the result of diagnostic runs, where we tested the variance and mean decay, which correspond to assumptions (3) and (1) from Theorem 1 with  $\beta = 1$  and  $\alpha = 1$  respectively (i.e. the slope of decay on the log-scale is  $-1$ ). This indicates, that Multilevel Monte Carlo achieves the same convergence rate as traditional Monte Carlo in terms of accuracy  $\varepsilon$ , but can offer computational savings, due to the fact that most of the samples are calculated for very small subset of minimal cut sets.

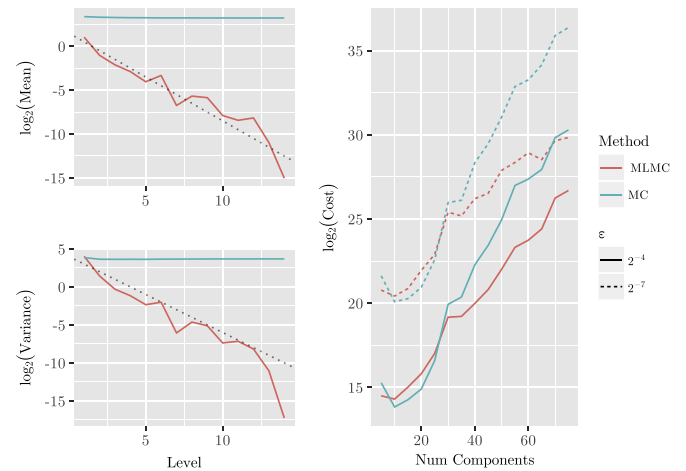
The right plot compares the differences in averaged costs for Multilevel Monte Carlo and standard Monte Carlo algorithms, which shows good savings even including the costs for initially selecting the cut sets for each level.

### 5.2.2. Shape parameter $k=1$

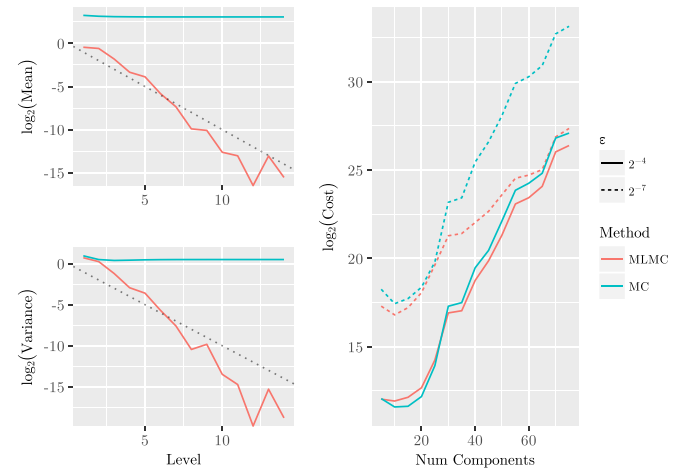
The test case with  $k=1$  (Fig. 3) gives us almost the same mean and variance decays along with computational gains as in the case with  $k=0.5$ . One can see that in the last levels we see even super linear decay for the mean and variance, which indicates that the cut sets being added at those levels have very weak impact on system lifetime compared to those already chosen, which indicates good performance for the level selection algorithm.

### 5.2.3. Shape parameter $k=3$

The case with  $k=3$  (Fig. 4) shows substantial savings for  $\varepsilon = 2^{-7}$ , as also seen in the previous examples, while still showing competitive results for  $\varepsilon = 2^{-4}$ . The reason the gains are more modest here is that,



**Fig. 3.** Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter  $k=1$  and uniformly distributed scale.



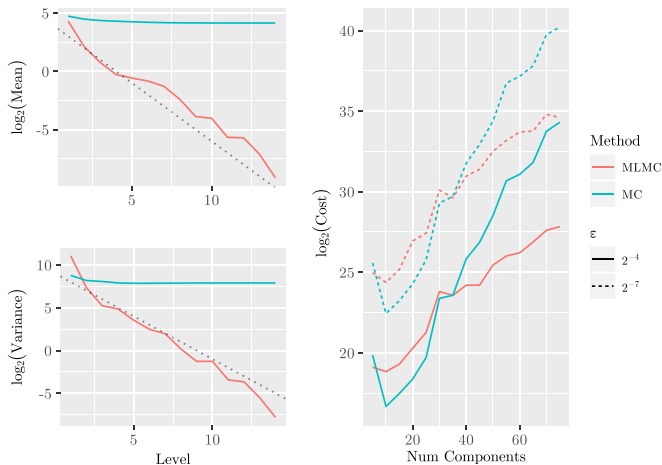
**Fig. 4.** Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter  $k=3$  and uniformly distributed scale.

as we had before, there is very small variance in the standard Monte Carlo and overall Multilevel Monte Carlo estimators, which puts more emphasis on the initial level selection costs which are not  $\varepsilon$  dependent, but are size dependent.

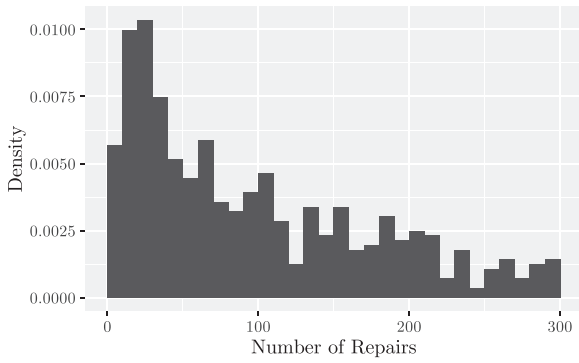
## 6. Numerical experiments, with repair process

To demonstrate the generality of the method and the substantial computational benefits available in more interesting scenarios, we consider the same 70 component system as generated in Section 5 with a repair process included. The components are again taken to have shape parameter  $k=0.5$  and uniformly distributed scale on  $[2, 10]$ , but now failed components are repaired according to an Exponentially distributed clock with rate  $\lambda = 0.05$ .

Note that the final failure time of all components which lead to system failure cannot be sampled simultaneously any more, because repairs may change the state of the system en-route to ultimate failure. Indeed, the computational complexity of sampling is substantially greater due to the need to simulate the stochastic process of failure and repair, repeatedly testing after each system state change whether the system is still functional. Consequently, obtaining a single Monte Carlo sample may now require many passes over the collection of cut sets and moreover there is additional randomness in the runtime to simulate a single system failure time.



**Fig. 2.** Left: Diagnostic tests for the largest considered system; Right: cost gains for nested randomly grown systems from 5 to 75 components, with Weibull distribution having shape parameter  $k=0.5$  and uniformly distributed scale.



**Fig. 5.** A standard Monte Carlo estimate of the distribution of number of repairs before system failure in the example system. Figure truncated at 300 repairs.

However, the Multilevel paradigm still applies and, as will be seen, even performs marginally better in this more complex scenario.

### 6.1. Repair process

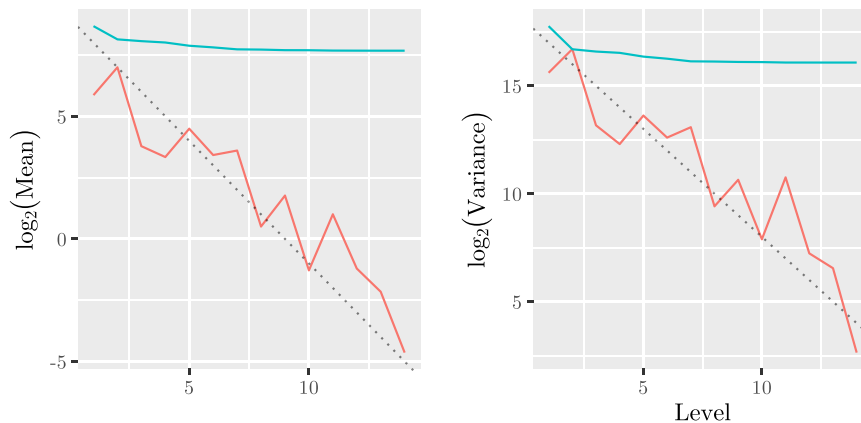
The repair process is taken to be Exponential ( $\lambda = 0.05$ ). Some full standard Monte Carlo runs show that this leads to a highly random number of repairs over the lifetime of the system as depicted in Fig. 5. Note that this is not so much chosen to be representative of any real system, but is in fact faster repair than might be expected in order to increase the difficulty of the problem.

### 6.2. Level selection for repairable systems

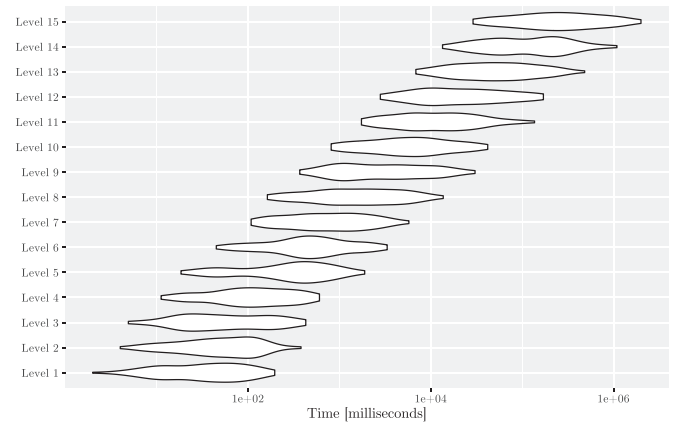
The procedure described for non-repairable systems can be adapted to this case. The only minor adjustment required is to the level selection algorithm as described in Section 4.2.1.

Recall that the level selection procedure first involves determining a failure time for *all* cut sets. In the non-repairable case, this was simply a case of computing Eq. (5) for a fixed collection of component simulations  $t_c^{(j)}$ . However, the stochastic process of failure and repair means that  $t_c^{(j)}$  depends on the cut set  $C_i$  which causes failure, so that in principle for each  $j$  the stochastic process should be simulated continuously until all cut sets have failed at least once, with the first failure time being recorded as  $\eta_i$  for each  $C_i \in C$ . As such, very rare failure modes may result in essentially unbounded compute cost.

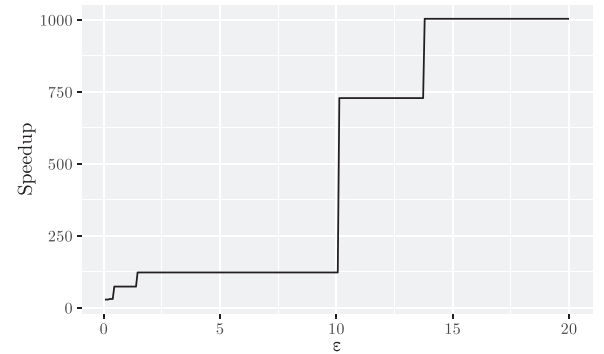
To address this, we propose simulation of the full stochastic process of failure and repair until the *first* instance of failure due to a cut set. We then simulate the conditional failure time of the still functioning components given survival to this time *without* further repairs taking



**Fig. 6.** Diagnostic plots for the repairable system example.



**Fig. 7.** Actual runtimes to perform a single sample on each level for the repairable example. Note the log-scaled time axis. Density estimates composed from 108 replicates. Timings are for a single core of a c4.8xlarge Amazon EC2 compute instance using the AMI from [1].



**Fig. 8.** Relative speedup of MLMC versus standard Monte Carlo to achieve estimation of expected lifetime within an accuracy of  $\epsilon$ .

place. Note that the exact behaviour beyond the initial cut set failure is therefore deemed less important: our primary goal is to establish cut sets which fail early, so exhaustive simulation is redundant. Clearly, these simulations cannot be used in the final estimate in the way they could for the non-repairable case.

In every other way, the level selection algorithm is the exact analogue of that for the non-repairable case, with the objective being to ensure rapid decay of the mean of each level.



### 6.3. Results

#### 6.3.1. Diagnostics

Fig. 6 (left) shows a mean decay of order 1, which implies that the expected contribution to the system lifetime estimate halves compared to the contribution from the previous level. This strong decay, corresponding to  $\alpha = 1$  in Theorem 1, means that our sorting approach does capture the influence of the cut sets, thus allowing estimation of the failure time with reasonable accuracy, without spending computational effort on evaluating ‘non-contributing’ cut sets. Recall that the stronger the mean decay (i.e. the larger is the value  $\alpha$ ), the fewer levels we will need, thus the higher will be the computational gain.

Fig. 6 (right) shows the variance decay and indicates the quality of the coupling. As each level is more expensive to sample than the previous one, it is desirable to sample it less often, without destroying the overall variance. The parameter  $\beta$  quantifies this in a rigorous way, in our case having  $\beta \approx 1$ , which implies that the variance on each level is half that on the previous level, hence halving the number of samples required.

#### 6.3.2. Computational cost

Recall that the number of cut sets in a level was an accurate proxy for the computational cost in the non-repairable example (see p.10) and that  $\gamma = 1$  could then be targeted by doubling the number of cut sets in each level.

However, in the repairable case this is no longer so, because the stochastic process of failure and repair adds a random element to the simulation runtime before ‘system’ failure, with it also depending on the cut set collection under consideration. Therefore, Fig. 7 shows the distribution of empirical wall-clock runtimes to produce a single simulation for each level on a log scale, showing the desired growth in compute cost. Regressing  $\log_2(\text{mean runtime}) = a + \gamma \ell$  results in  $\gamma \approx 0.94$ . Crucially, this means that an exponential improvement in accuracy is achieved ( $\alpha = \beta = 1$ ), but with a little below exponential increase in cost ( $\gamma \approx 0.94$ ). This means MLMC actually provides marginally better performance gain in the repairable case than it did in the non-repairable case (where  $\alpha = \beta = \gamma = 1$ ).

When computing the cost in the repairable case, we can use the empirical mean compute time for level  $\ell$ ,  $\kappa_\ell$  say, instead of the number of cut sets in Eq. (4). Then, for a target accuracy  $\varepsilon$ , the speedup provided by MLMC is characterised by:

$$\text{Speedup} = \frac{\varepsilon^{-2} \text{Var}(\tau_i) \kappa_L}{\varepsilon^{-2} \left( \sum_{\ell=0}^{L_\varepsilon} \sqrt{\text{Var}(Y_\ell) \kappa_\ell} \right)^2}$$

where  $L_\varepsilon$  is the earliest level with mean less than  $\varepsilon$ . For varying  $\varepsilon$  this is depicted in Fig. 8. For coarse estimates, speedups of upto 1000 times can be achieved. Note that the expected system lifetime is  $\approx 205$  with high variance  $\approx 69,000$  — this was chosen as an extreme example to test MLMCs ability to handle very difficult estimation problems.

To put this in perspective, the time to achieve a Monte Carlo estimate to accuracy  $\varepsilon = 1.5$  would take about 141 days on a single CPU core (or nearly 4 days using all cores of the c4.8xlarge Amazon EC2 instance used for testing). Multilevel Monte Carlo would achieve the same accuracy in 1 day, 4 h on a single CPU core (or just 45 min using all cores of a c4.8xlarge).

## 7. Conclusion and future work

We have presented an exciting new application for the Multilevel paradigm for estimating the reliability of systems, which speeds up traditional Monte Carlo estimation of system lifetimes and provides a approach which can easily generalise to other reliability problems which involve cut (path) sets. We have demonstrated that the proposed approach to using MLMC in reliability problems achieves the strong mean and variance decay required to enable truncation of the number

of levels which must be simulated. This is a very desirable feature for MLMC, as the standard approach has to go through all the cut sets for each sample, regardless of target accuracy. Moreover, we have demonstrated that harder problems (such as repairable systems) achieve slightly greater gains through just below exponential growth in the cost of simulating levels ( $\gamma \approx 0.94$ ) while still having exponential mean and variance decay ( $\alpha = \beta = 1$ ).

Unlike classic MLMC implementations, where one considers different approximations of a certain stochastic process wherein all of them are biased, here we introduce approximations based on sorting the minimal cut sets in a special way, which are biased, but less costly to simulate. The numerical experiments show substantial savings for large systems and are promising for further study of reliability and structural optimisation. Indeed, the ease of extending from non-repairable to a repairable setting shows the flexibility of the approach and we anticipate it would be likewise straight-forward to incorporate shock and stress processes in a similar manner, benefitting ever more greatly from the acceleration offered by MLMC. Our own future work will include the extension of our approach in the spirit of [13], and expanding the applicability of the Multilevel paradigm to other algorithms established in the reliability community.

## Acknowledgements

The authors would like to thank four reviewers for comments which greatly improved the presentation of this paper.

Louis Aslett is supported by the i-like programme grant (EPSRC grant reference number EP/K014463/1 <http://www.i-like.org.uk>). Tigran Nagapetyan and Sebastian Vollmer received funding from EPSRC Grant EP/N000188/1.

## References

- [1] Aslett LJM. RStudio AMIs for Amazon EC2 cloud computing, 2016. AMI ID ami-8fe18f98, ([http://www.louisaslett.com/RStudio\\_AMI/](http://www.louisaslett.com/RStudio_AMI/)).
- [2] Aslett LJM, Coolen FPA, Wilson SP. Bayesian inference for reliability of systems and networks using the survival signature. *Risk Anal* 2015;35(9):1640–51.
- [3] Aslett LJM, Giles MB, Nagapetyan T, Vollmer SJ. mlmc: Tools for Multilevel Monte Carlo, 2016. R package, (<http://cran.r-project.org/package=mlmc>).
- [4] Barlow RE, Proschan F. Mathematical theory of reliability. Wiley; 1965.
- [5] Barlow RE, Proschan F. Statistical Theory of Reliability and Life Testing. To Begin With Press, 1981.
- [6] Butterworth RW. A set theoretic treatment of coherent systems. *SIAM J Appl Math* 1972;22(4):590–8.
- [7] Chiacchio F, D’Urso D, Manno G, Compagno L. Stochastic hybrid automaton model of a multi-state system with aging: reliability assessment and design consequences. *Reliab Eng Syst Saf* 2016;149:1–13.
- [8] Coolen FPA, Coolen-Maturi T. Generalizing the signature to systems with multiple types of components. In Zamojski W, Mazurkiewicz J, Sugier J, Walkowiak T, Kacprzyk J. editors, *Complex Systems and Dependability*, volume 170 of *Advances in Intelligent and Soft Computing*, p. 115–130. Springer Berlin Heidelberg, 2012.
- [9] Coolen FPA, Coolen-Maturi T, Al-Nefaie AH. Nonparametric predictive inference for system reliability using the survival signature. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 228(5): 2014; p. 437–448.
- [10] Giles MB. Multilevel Monte Carlo path simulation. *Oper Res* 2008;56(3):607–17.
- [11] Giles MB. Matlab code for multilevel Monte Carlo computations. (<http://people.maths.ox.ac.uk/gilesm/acta/>), 2014.
- [12] Giles MB. Multilevel Monte Carlo methods. *Acta Numer* 2015;24:259–328.
- [13] Giles MB, Nagapetyan T, Ritter K. Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):2015; p. 267–295.
- [14] Heinrich S. Monte Carlo complexity of global solution of integral equations. *J Complex* 1998;14(2):151–75.
- [15] Jun CH, Ross SM. System reliability by simulation: random hazards versus importance sampling. *Probab Eng Inf Sci* 1992;6(01):119–26.
- [16] Kebaier A. Statistical Romberg extrapolation: omberg extrapolation: a new variance reduction method and applications to options pricing. *Ann Appl Probab* 2005;14(4):2681–705.
- [17] Kloeden PE, Platen E. Numerical solution of stochastic differential equations, 23. Springer; 1992.
- [18] Kocher S, Mukerjee H, Samaniego FJ. The “signature” of a coherent system and its application to comparisons among systems. *Nav Res Logist* 1999;46(5):507–23.
- [19] Manno G, Chiacchio F, Compagno L, D’Urso D, Trapani N. Conception of repairable dynamic fault trees and resolution by the use of raatss, a matlab toolbox based on the ats formalism. *Reliab Eng Syst Saf* 2014;121:250–62.

- [20] Merle G, Roussel J-M, Lesage J-J. Algebraic determination of the structure function of dynamic fault trees. *Reliab Eng Syst Saf* 2010;96(2):267–77.
- [21] Merle G, Roussel J-M, Lesage J-J. Quantitative analysis of dynamic fault trees based on the structure function. *Qual Reliab Eng Int* 2014;30(1):143–56.
- [22] Rai S, Veeraraghavan M, Trivedi KS. A survey of efficient reliability computation using disjoint products approach. *Networks* 1995;25(3):147–63.
- [23] Samaniego FJ. On closure of the IFR class under formation of coherent systems. *IEEE Trans Reliab* 1985;R-34(1):69–72.
- [24] Shooman ML. Reliability of computer systems and networks: fault tolerance, analysis, and design. John Wiley & Sons; 2003.
- [25] Xing L, Amari SV. Binary Decision Diagrams and extensions for system reliability analysis. John Wiley & Sons; 2015.
- [26] Xing L, Shrestha A, Dai Y. Exact combinatorial reliability analysis of dynamic systems with sequence-dependent failures. *Reliab Eng Syst Saf* 2011;96(10):1375–85.
- [27] Yeh WC. An improved sum-of-disjoint-products technique for symbolic multi-state flow network reliability. *IEEE Trans Reliab* 2015;64(4):1185–93.
- [28] Olexandr Y. An improved monte carlo method in fault tree analysis. In *Reliability and Maintainability Symposium (RAMS), 2010 Proceedings-Annual*, IEEE, 2010; p. 1–5.
- [29] Zang X, Wang D, Sun H, Trivedi KS. A BDD-based algorithm for analysis of multistate systems with multistate components. *IEEE Trans Comput* 2003;52(12):1608–18.
- [30] Zuo MJ, Tian Z, Huang HZ. An efficient method for reliability evaluation of multistate networks given all minimal path vectors. *IIE Trans* 2007;39(8):811–7.